hiprom technologies

1756HP-GSM

USER MANUAL

Rev 2.2.4 – April 2008

Table of Contents

CHAPTER 1 INTRODUCTION

Hiprom presents the 1756HP-GSM interface module. The 1756HP-GSM is an integrated GSM product that slots into the Allen Bradley ControlLogix PLC system. The 1756HP-GSM (master) module utilizes two forms of GSM communication to transmit information to/from the PLC:

1. SMS (Short Message Service).
2. GPRS (General Packet Radio Service)

The module fits directly into any single slot within a ControlLogix System and is configured in RSLogix5000 using the Generic Profile. A SMA connector located on the underside of the module provides an interface to the antenna supplied.

This document serves to describe the functionality, installation, configuration and use of the module.

1. SMS

The **Short Message Service (SMS)**, often called text messaging, is a means of sending short messages to and from GSM modules.

2. GPRS

The **General Packet Radio Service (GPRS)** is a service that allows information to be sent and received across a mobile telephone network. It supplements today's Circuit Switched Data and Short Message Service.

GPRS is <u>not</u> related to GPS (Global Positioning System).

CHAPTER 2 MODULE ACCESSORIES

Each 1756HP-GSM package includes the following components:

- **1756HP-GSM module**
- **GSM Antenna with patch lead**

CHAPTER 3 MODULE OPERATION

The 1756HP-GSM module is designed to operate within the Allen-Bradley ControlLogix PLC system. All power required for the module's operation is derived from the 1756 backplane.

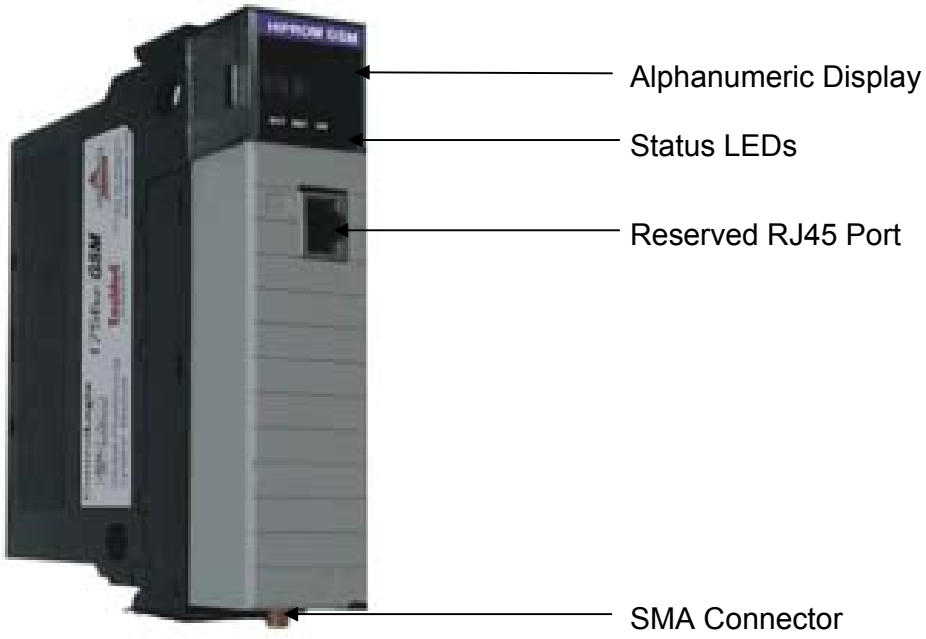


Figure 3.1 : *1756HP-GSM Layout*

The current status of the module is conveyed to the user by means of the 3 bi-color Status LED's and the alphanumeric LED display. Appendix A details the various states of the LED's and messages of alphanumeric display.

The following information is available to the user directly across the backplane by means of a scheduled connection:

- Status of the GSM module
- Signal Strength (Expressed in percentage %)
- Indication of new SMS and size of new SMS

The SMA connector located on the bottom of the module provides an external connection for the GSM antenna. The module is supplied with the respective patch cable.

Other information is also available with respect to the GPRS functionality:

- Local IP Address of SIM.
- Indication of received CSpeak / GPRS packet data.

CHAPTER 4 INSTALLING THE MODULE

The module is equipped with RIUP (Removal and Insertion Under Power) circuitry enabling the module to be installed or removed from the chassis while power is applied.

CHAPTER 5 CONFIGURING THE MODULE

A direct connection between the controller and the 1756HP-GSM module is required to transfer I/O data to and from the module.

Establishing the Direct Connection

This section describes the procedures necessary to configure the 1756HP-GSM module within the ControlLogix system. Each 1756HP-GSM module must be owned by a single ControlLogix controller.

The 1756 Generic Module is used in RSLogix5000 to configure the module. The configuration of the module is detailed in the tables below.

Module Configuration:

Data Format		
CommFormat	Data –**DINT**	
Connection parameters		
Description	Instance	Size
Input	1	63(32 Bit)
Output	2	50 (32 Bit)
Configuration	4	400 (8 Bit)
RPI		
RPI	100 msec	

***Table 5.1**: 1756HP-GSM connection parameters.*

The steps required to add a new 1756HP-GSM module are detailed below.

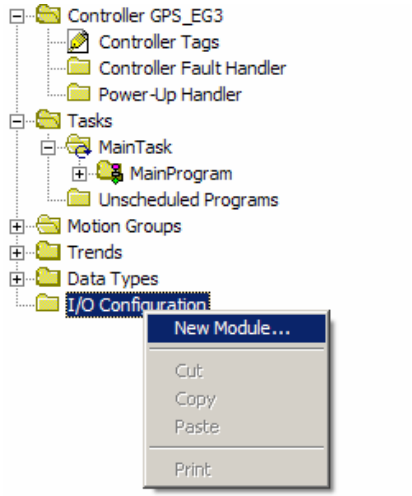


Figure 5.1 : *Right-click on I/O Configuration and select New*

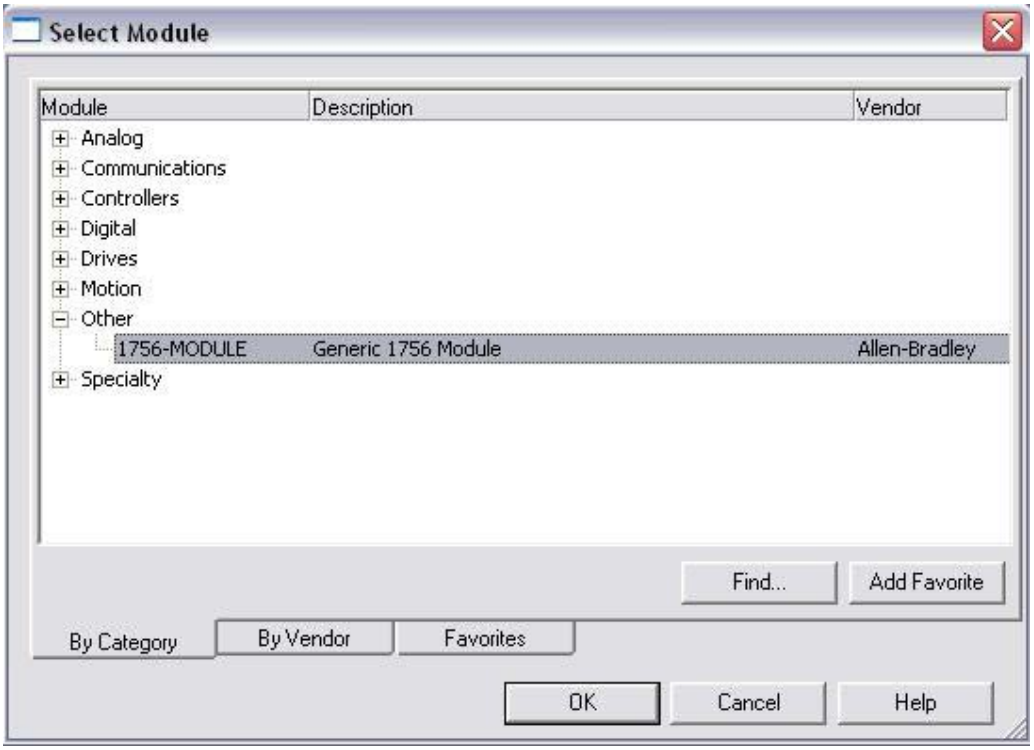


Figure 5.2: *Select Generic 1756 Module (1756-MODULE)*

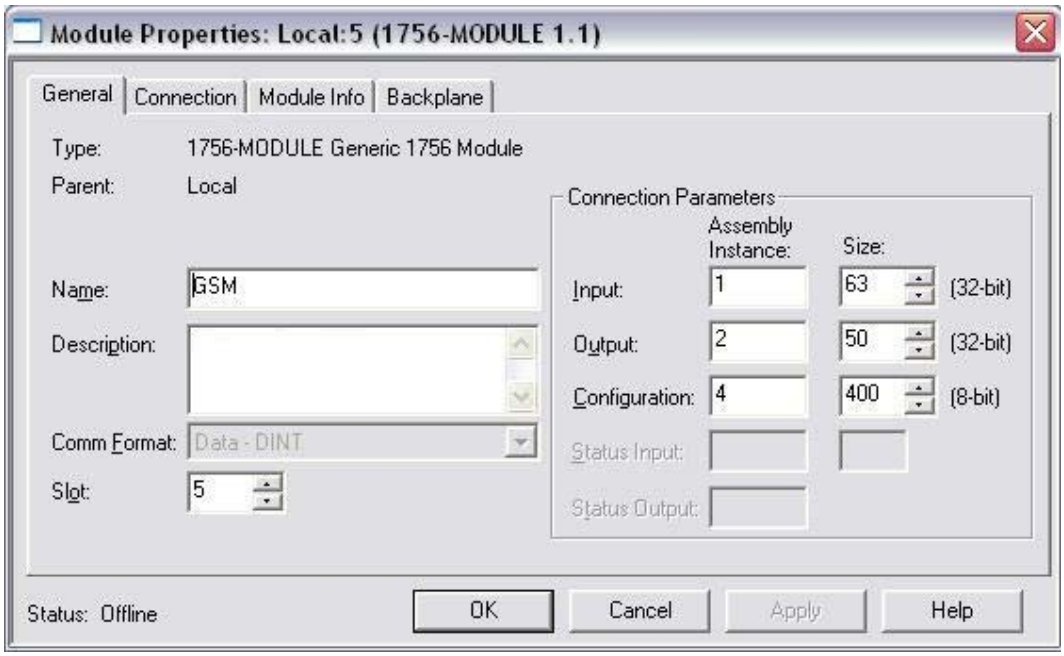


Figure 5.3: *Configure module's parameters*

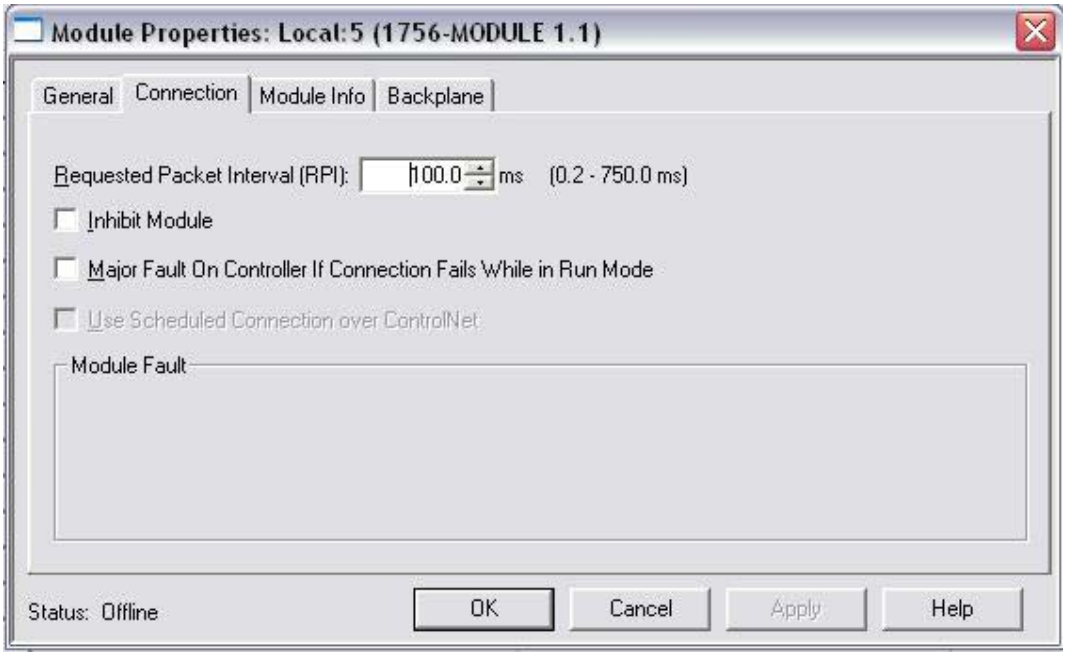
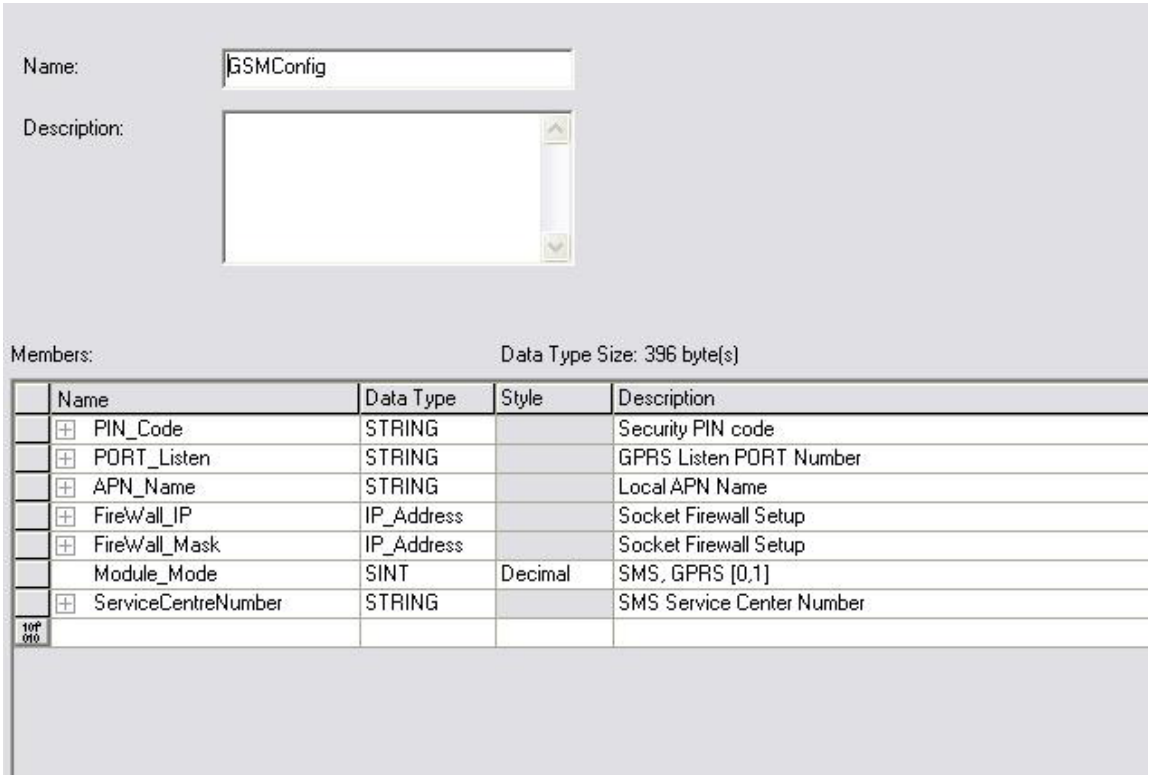


Figure 5.4: *Configure module's RPI (Requested Packet Interval)*

The configuration data is defined in a UDT of type ***GSMConfig*** defined as follows:



Name	Data Type	Style	Description
PIN_Code	STRING		Security PIN code
PORT_Listen	STRING		GPRS Listen PORT Number
APN_Name	STRING		Local APN Name
FireWall_IP	IP_Address		Socket Firewall Setup
FireWall_Mask	IP_Address		Socket Firewall Setup
Module_Mode	SINT	Decimal	SMS, GPRS [0,1]
ServiceCentreNumber	STRING		SMS Service Center Number

Figure 5.5: *GSMConfig Configuration UDT*

Figure 5.6: *Configuration ladder code*

In the above figure, GSMConfig is a user data type and is copied to the module in this manner.

. The configuration data specifies the following:

- The PIN number of the SIM card
- The GPRS Listening Port Number
- The GPRS APN Name
- The GPRS Firewall IP
- The GPRS Firewall mask
- The GSM *Module Mode*
- The SMS Service Center Number

The different GSM *Module types* are:

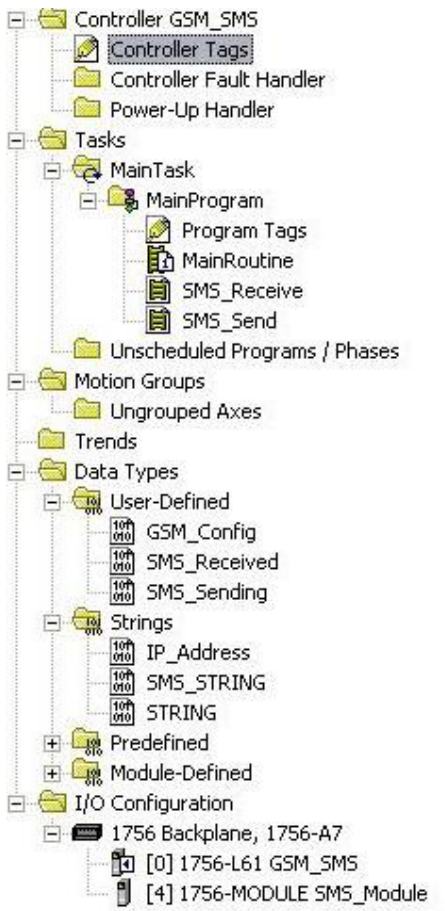
0 – Standard SMS Module (GSM, GPRS disabled)
1 – GPRS Module (GPRS, SMS also enabled)

Note: The Configuration is ONLY read when the card initializes itself. If any data is changed, the card must be reset to enable the changes. The mode only needs to be set once.

Very Important: When booting up the 1756HP-GSM module, enter the correct PIN number in the configuration.

Booting up with the incorrect PIN **3 times** will result in the SIM card being locked requiring a PUK security code.

Below is an example of what the RSLogix Drop-down tree menu might look like:



Very Important: When booting up the 1756HP-GSM module, enter the correct PIN number in the configuration.

Booting up with the incorrect PIN **3 times** will result in the SIM card being locked requiring a PUK security code.

The Configuration data is as follows:

GSMConfig	{...}	{...}		GSMConfig	
GSMConfig.PIN_Code	'4743'	{...}		STRING	Security PIN code
GSMConfig.PORT_Listen	'5050'	{...}		STRING	GPRS Listen PORT Number
GSMConfig.APN_Name	'www.hiprom.co.za'	{...}		STRING	Local APN Name
GSMConfig.FireWall_IP	'196.135.148.1'	{...}		IP_Address	Socket Firewall Setup
GSMConfig.FireWall_Mask	'255.255.255.0'	{...}		IP_Address	Socket Firewall Setup
GSMConfig.Module_Mode	1		Decimal	SINT	SMS, GPRS [0,1]
GSMConfig.ServiceCentreNumber	'+27829119'	{...}		STRING	SMS Service Center Number

Figure 5.7: *GSMConfig Configuration*

Should the SIM being used with the module require a PIN to be entered the Configuration data must be setup. For example should the PIN be “4743”, the above data would be entered.

Once the module has been inserted into the rack and powered, the card will evaluate whether a PIN for the SIM is required. If a PIN is required the card will load the PIN configured in the Configuration Image.

Once the SIM has been initialized correctly the module will attempt to establish connection with the GSM network. The Red GSM status LED (See Appendix A) should be on, and the Green GSM Registration LED will be flashing quickly. The module will take approximately 1 minute to configure and establish a connection with the GSM network. Once it has successfully connected the Green GSM Registration Status LED will begin flashing slowly. The module is now ready to send and receive SMSs or transmit/receive GPRS data.

The *GSM_Status* UDT is shown below as well as an example of a Tag with its corresponding example values.

Name: GSM_Status

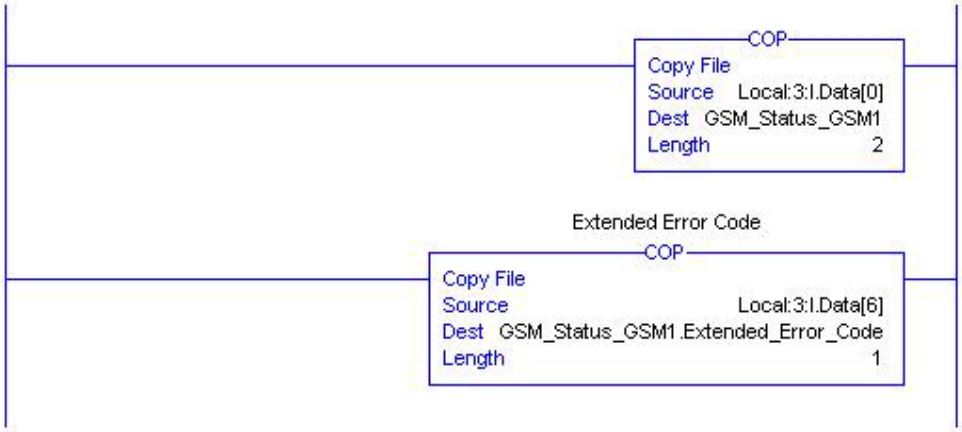
Description:

Members: Data Type Size: 12 byte(s)

Name	Data Type	Style	Description
Online_OR_Busy	BOOL	Decimal	Online(1) / Busy (0)
New_SMS_Received	BOOL	Decimal	SMS Has Arrived
PIN_Required_On_Boot	BOOL	Decimal	PIN Code Request is ON (1) Off (0)
PUK_Required_SIMFail	BOOL	Decimal	PUK Code is reuiqued to unlock SIM
SIM_Error	BOOL	Decimal	SIM error Has occured
GPRS_Error	BOOL	Decimal	GPRS error has occured
GPRS_Rx	BOOL	Decimal	GPRS Packet Recieved
GPRS_Listen_Mode	BOOL	Decimal	ListenOn(1)
Reserved_1	BOOL	Decimal	Reserved
Reserved_2	BOOL	Decimal	Reserved
Module_Mode	BOOL	Decimal	SMS(0) / GPRS+SMS (1)
Reserved_3	BOOL	Decimal	Reserved
Reserved_4	BOOL	Decimal	Reserved
Reserved_5	BOOL	Decimal	Reserved
Reserved_6	BOOL	Decimal	Reserved
Reserved_7	BOOL	Decimal	Reserved
ErrorCode	SINT	Decimal	Error Code
Reserved_8	SINT	Decimal	Reserved
Signal_Strength	DINT	Decimal	GSM Signal Strength
Extended_Error_Code	DINT	Decimal	Extended Error Code

Name	Value	Force	Style	Data Type	Description
GSM_Status	{...}	{...}		GSM_Status	
GSM_Status.Online_OR_Busy	1		Decimal	BOOL	Online(1) / Busy (0)
GSM_Status.New_SMS_Received	0		Decimal	BOOL	SMS Has Arrived
GSM_Status.PIN_Required_On_Boot	1		Decimal	BOOL	PIN Code Request is ON (1) Off (0)
GSM_Status.PUK_Required_SIMFail	0		Decimal	BOOL	PUK Code is reuiqued to unlock SIM
GSM_Status.SIM_Error	0		Decimal	BOOL	SIM error Has occured
GSM_Status.GPRS_Error	0		Decimal	BOOL	GPRS error has occured
GSM_Status.GPRS_Rx	0		Decimal	BOOL	GPRS Packet Recieved
GSM_Status.GPRS_Listen_Mode	0		Decimal	BOOL	ListenOn(1)
GSM_Status.Reserved_1	0		Decimal	BOOL	Reserved
GSM_Status.Reserved_2	0		Decimal	BOOL	Reserved
GSM_Status.Module_Mode	0		Decimal	BOOL	SMS(0) / GPRS+SMS (1)
GSM_Status.Reserved_3	0		Decimal	BOOL	Reserved
GSM_Status.Reserved_4	0		Decimal	BOOL	Reserved
GSM_Status.Reserved_5	0		Decimal	BOOL	Reserved
GSM_Status.Reserved_6	0		Decimal	BOOL	Reserved
GSM_Status.Reserved_7	0		Decimal	BOOL	Reserved
GSM_Status.ErrorCode	0		Decimal	SINT	Error Code
GSM_Status.Reserved_8	0		Decimal	SINT	Reserved
GSM_Status.Signal_Strength	70		Decimal	DINT	GSM Signal Strength
GSM_Status.Extended_Error_Code	0		Decimal	DINT	Extended Error Code

The following ladder is added to the program to copy the status over from the card's Input Image.



CHAPTER 6 I/O ADDRESS MAP

The input and output image of the 1756HP-GSM module is defined in the following sections.

Input Image (32bit words)

Word	31	30	29	28	27	26	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved							Error Code (0 - 255)								Reserved	Reserved	Reserved	Reserved	Reserved	SMSmode(0)/GPRSmode(1)	Reserved	Reserved	GPRS Listen Mode	CSpeak/GPRS RX	GPRS Error	SIM ERROR	PUK Required	PIN Required	New SMS Received	Online [1] / Busy [0]
1	Signal Strength																														
2	Received SMS Count																														
3	Received SMS Text Size																														
4	Local IP Address(Hex) (Byte 4… Byte 1) (GPRS)																														
5	Last Connection IP FROM(Hex) (Byte 4 … Byte 1) (GPRS)																														
6	Ext Error Code																														
7	ContextNotOpenedCnt [H-int]CRC_ErrCnt[L-int]																														
8	Num_Activation_failedCnt [H-int] Num_IP_receivedCnt[L-int]																														
9	Internal_RecCnt[L-int]																														
10	Reserved																														
11	Reserved																														
12	RX Packet TNS_Num (H -int) GRPS Data Size to Follow(L-int)																														
13 … 62	GPRS Data (50 DINTs Max)																														

Table 6.1: *Connected Input Image*

Output Image

Words 0...50 DINTs

Word	Description
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
….	Etc…

Input Image Description

Field/Value	Description	Location	Type
Online [1] / Busy [0]	**Indicates that the module has successfully initialized and that the module is connected to the GSM network – indicates if module is busy**	**Local:s:I.Data[0].0**	BIT
New SMS Received	**Indicates new SMS received:** 0 = No new SMS received 1 = New SMS received	**Local:s:I.Data[0].1**	BIT
PIN required	**Indicates that the SIM card the module uses requires a PIN number to be entered. Please refer to the configuration of the module** 0 = No PIN required 1 = PIN required	**Local:s:I.Data[0].2**	BIT
PUK required	**Indicates that the SIM card the module uses requires a PUK number to be entered. The SIM needs to be removed from the module and configured.**	**Local:s:I.Data[0].3**	BIT
SIM Error	**There is a problem with the module's SIM card**	**Local:s:I.Data[0].4**	BIT
GPRS Error	**There is a GPRS specific error.**	**Local:s:I.Data[0].5**	BIT
CSpeak/GPRS RX	**Some form of CSpeak/GPRS packet has been received.** 0 = GPRS request has been issued. 1 = New CSpeak/GPRS response has been received.	**Local:s:I.Data[0].6**	BIT
Error Code	**Should the module experience an error, an error code will be inserted.**	**Local:s:I.Data[0].8..15**	SINT
Signal Strength	**The strength of the signal expressed in percentage (%)**	**Local:s:I.Data[1]**	DINT
Received SMS Count	**Number of SMS received.** This Field will increment each time a new SMS is received	**Local:s:I.Data[2]**	DINT
Received SMS Text Size	**Size of Received SMS Text** This will indicate the size (number of characters) of the new SMS received.	**Local:s:I.Data[3]**	DINT
Ext Error Code	**Extended Error Code**	**Local:s:I.Data[6]**	DINT
Local IP Address	**Local IP address of Module** This is the local IP address assigned to the SIM card by the network supplier.	**Local:s:I.Data[4]**	DINT

Last Connection IP FROM	**Shows the last IP address that has contacted this module.**	**Local:s:I.Data[5]**	DINT
CRC_ErrCnt [L-int]	**Number of CRC errors received.**	**Local:s:I.Data[7]**	INT
ContextNotOpenedCnt [H-int]	**Number of GPRS contexts that have failed.**	**Local:s:I.Data[7]**	INT
IP_receivedCnt[L-int]	**Number of times IP has been successfully returned from network.**	**Local:s:I.Data[8]**	INT
Activation_failedCnt [H-int]	**Number of times Context activation has failed.**	**Local:s:I.Data[8]**	INT
Internal_RecCnt[L-int]	**Number of Internal Network recoveries.**	**Local:s:I.Data[9]**	INT
RX Packet TNS_Num (H -int)	**Number that increments when Data Packet Arrives – Can be used to track Function 3 Data packet Arrival.**	**Local:s:I.Data[12]**	INT
GRPS Data Size to Follow	**The number of INTs that have been received from another GSM module.**	**Local:s:I.Data[12]**	INT
GPRS Data	**Physical data that has been received. Data is received. High_INT… Low_INT (16#54ae_6542)**	**Local:s:I.Data[13].... Local:s:I.Data[62]**	DINT

Table 6.2: *Input Image description*

CHAPTER 7 SENDING AND RECEIVING A SMS

Sending an SMS

SMS_Send Routine

An SMS is sent by issuing a custom CIP message to the module. The ladder below is the recommended procedure to send the SMS using a timer.

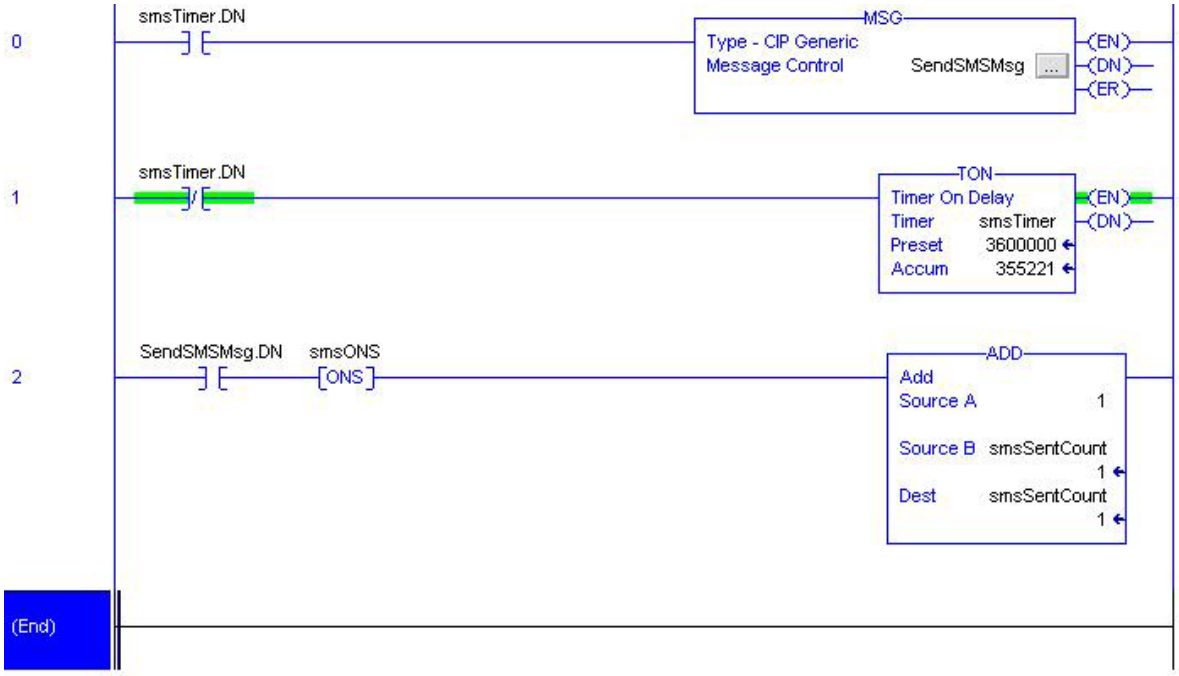


Figure 7.1: *Send SMS Ladder Example*

Field	Value
Message Type	CIP Generic
Service Type	Custom
Service Code	0x32
Class	0x71
Instance	0x01
Attribute	0x01
Source Length	184 bytes
Destination Element	Destination tag for reply data

The destination element is a dummy array and named “Response” or “SMS_Response”.

The message (***SendSMSMsg***) needs to be configured as follows:

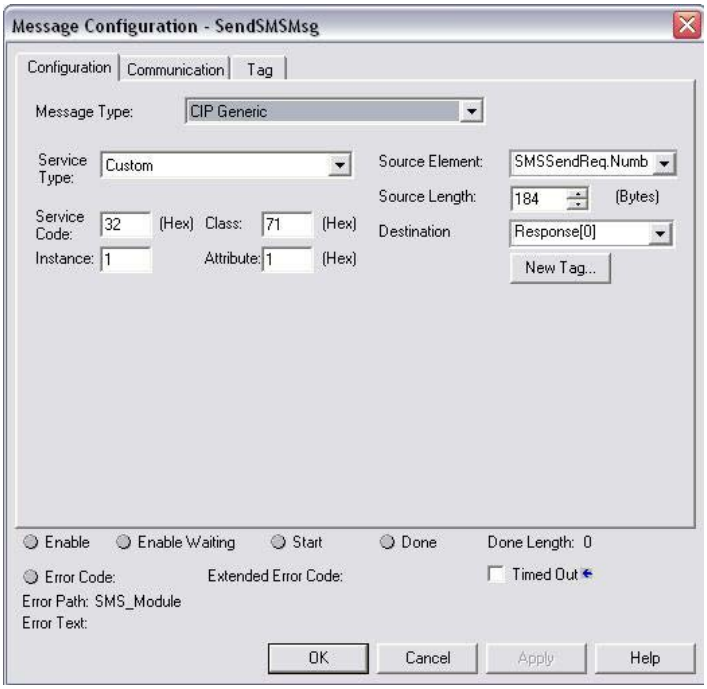


Figure 7.2: *Send SMS Message Configuration*

The source tag/element ***"SMSSendReq"*** is a UDT (User Data Type) of type ***SMS_Sending***, which is detailed below.

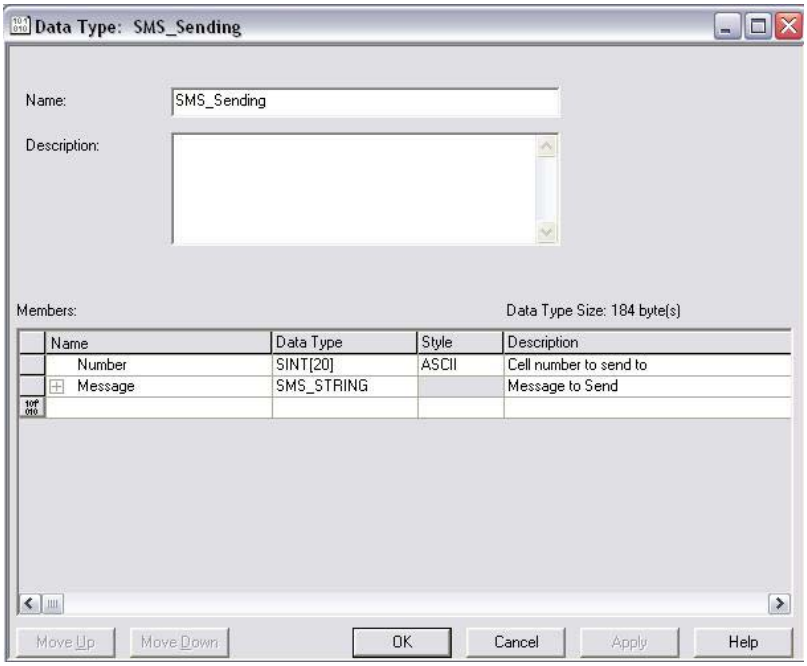


Figure 7.3.1: *SMS_Sending UDT*

The SMS_STRING data type is defined as follows in the following diagram:

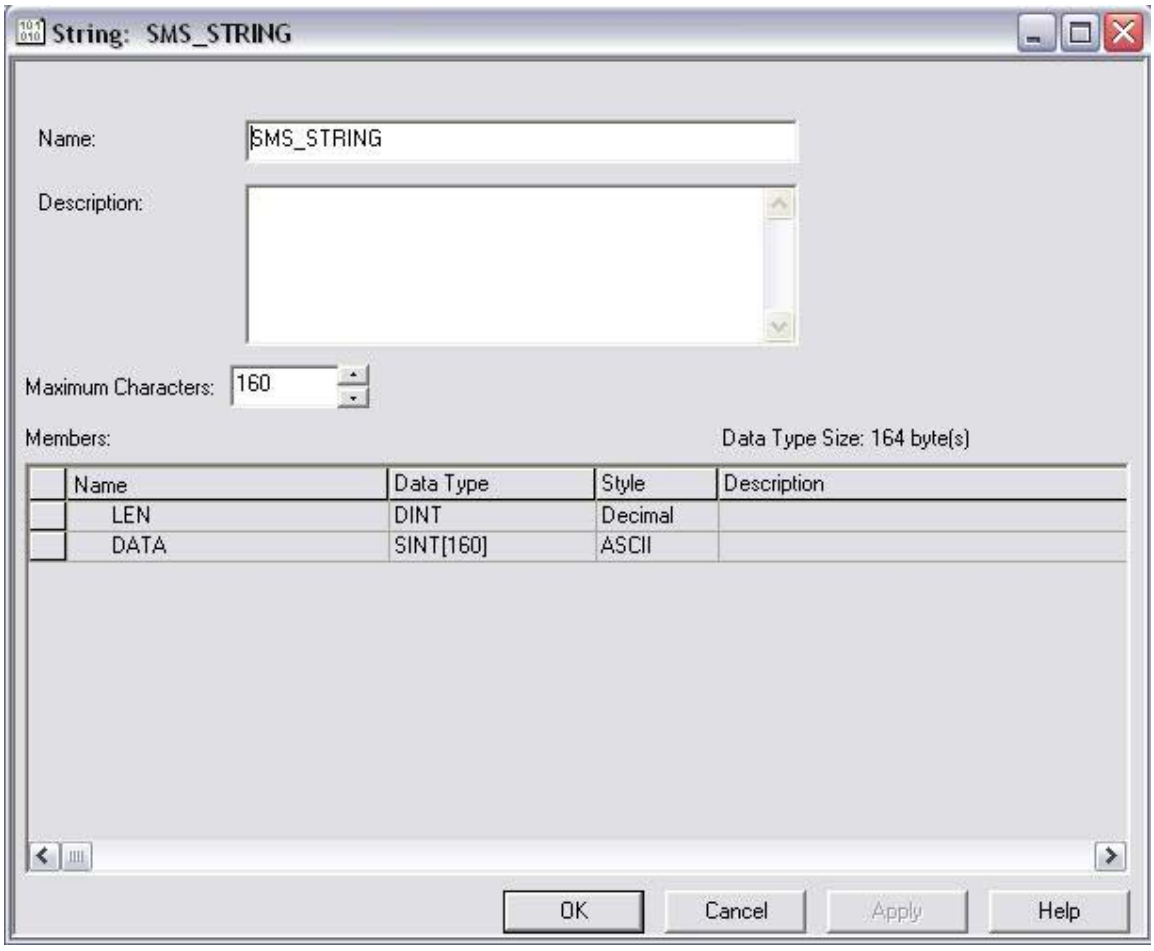


Figure 7.3.2: *SMS_STRING UDT*

Example of sending an SMS

Let's assume we would like to send an SMS to "0851234567" with the text "Controller GSM: Trip on 12 March 2006 at 10:00, Line 2". The source Tag ***"SMSSendReq"*** will be configured as follows:

SMSSendReq	{...}	{...}		SMS_Sending	
SMSSendReq.Number	{...}	{...}	ASCII	SINT[20]	Cell number to send to
SMSSendReq.Number[0]	'0'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[1]	'8'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[2]	'5'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[3]	'1'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[4]	'2'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[5]	'3'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[6]	'4'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[7]	'5'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[8]	'6'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[9]	'7'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[10]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[11]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[12]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[13]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[14]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[15]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[16]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[17]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[18]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Number[19]	'$00'		ASCII	SINT	Cell number to send to
SMSSendReq.Message	'Controller GSM: Trip on 12 March 2006 ...	{...}		SMS_STRING	Message to Send

Figure 7.4: *SMS Message Setup*

SMSSendReq.Message will be:

Figure 7.5: *SMS Message*

Note: The module is only capable of sending and receiving text of up to ***160*** characters.
Note: CELL NUMBER IS IN ASCII format.

Receiving an SMS

In the event of a new SMS being received, the module will:

1) Set bit 1 (***"New SMS Received"***) of Local:s:I.Data[0]
2) Increment Local:s:I.Data[2] (***"Received SMS Count"***)
3) Set Local:s:I.Data[3] (***"Received SMS Text Size Count"***) to the character size of the new SMS (refer to section 6.2).

Field	Value
Message Type	CIP Generic
Service Type	Custom
Service Code	0x32
Class	0x71
Instance	0x01
Attribute	0x02
Source Length	0 bytes
Destination Element	Destination tag for reply data

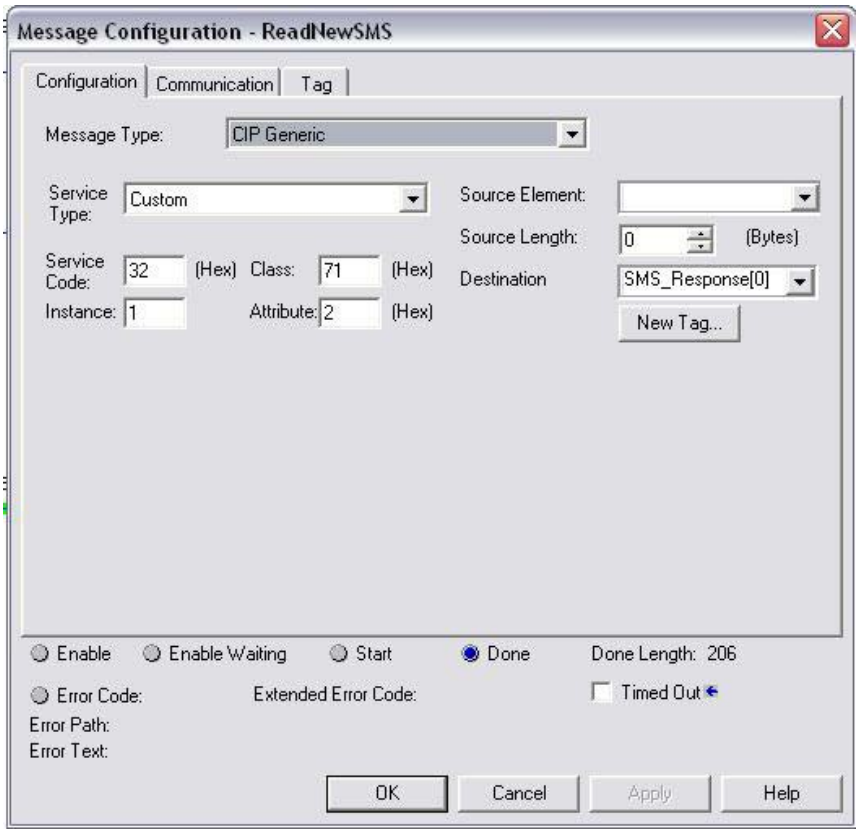


Figure 7.6: *Read SMS Message*

The response array holds the following data once a SMS has been received:

Name		Force Mask	Style	Data Type
Response	{...}	{...}	ASCII	INT[70]
Response[0]	'2+'		ASCII	INT
Response[1]	'88'		ASCII	INT
Response[2]	'28'		ASCII	INT
Response[3]	'78'		ASCII	INT
Response[4]	'06'		ASCII	INT
Response[5]	'23'		ASCII	INT
Response[6]	'$00$00'		ASCII	INT
Response[7]	'$00$00'		ASCII	INT
Response[8]	'$00$00'		ASCII	INT
Response[9]	'$00$00'		ASCII	INT
Response[10]	'$00$00'		ASCII	INT
Response[11]	'70'		ASCII	INT
Response[12]	'0/'		ASCII	INT
Response[13]	'/9'		ASCII	INT
Response[14]	'50'		ASCII	INT
Response[15]	'0,'		ASCII	INT
Response[16]	':7'		ASCII	INT
Response[17]	'62'		ASCII	INT
Response[18]	'1:'		ASCII	INT
Response[19]	'+1'		ASCII	INT
Response[20]	'80'		ASCII	INT
Response[21]	'eH'		ASCII	INT
Response[22]	'll'		ASCII	INT
Response[23]	' o'		ASCII	INT
Response[24]	'$00$00'		ASCII	INT
Response[25]	'$00$00'		ASCII	INT
Response[26]	'$00$00'		ASCII	INT
Response[27]	'$00$00'		ASCII	INT
Response[28]	'$00$00'		ASCII	INT

Figure 7.7: *Response Array*

Note: The Response array loads its INTEGER values low-byte first.

Note: Message received in this example, above figure, is “Hello”.
Response[21] … Response[23].

Note: This is viewed in ASCII style.

The following array positions are important:

Note: Response[26] denotes array position “26”

Array position	Description
0	Start of Cell Number of Sending SMS SIM card
11	Start of Date and Time
21	Start of Message Received

Table 7.1: *Response Array positions*

Refer to Figure 7.7 when using this table.

Note: The module is only capable of sending and receiving text of up to ***160*** characters i.e: Largest message Response[21]…Response[100]

SMS_Receive Routine

The example code illustrates the recommended procedure to read a new SMS. Please note that once an SMS has been read the SMS will be deleted from the SIM card in the module to ensure that the card will not run out of memory. The ladder example code below thus stores all the received SMS in an array (SMS_List) of data type **SMS_Received** (as detailed in Figure 7.8.2).

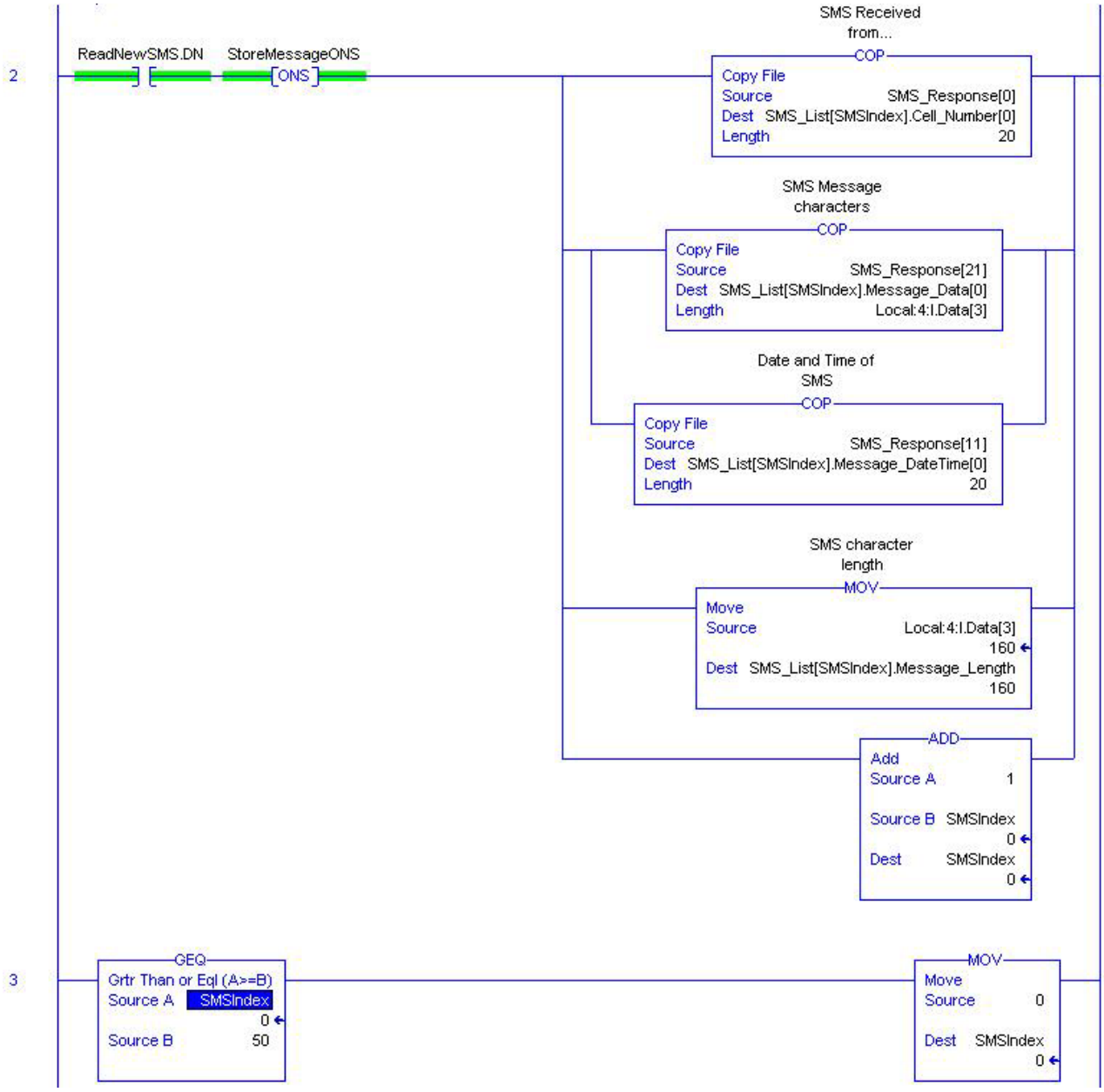


Figure 7.8.1: *Processing Response Array*

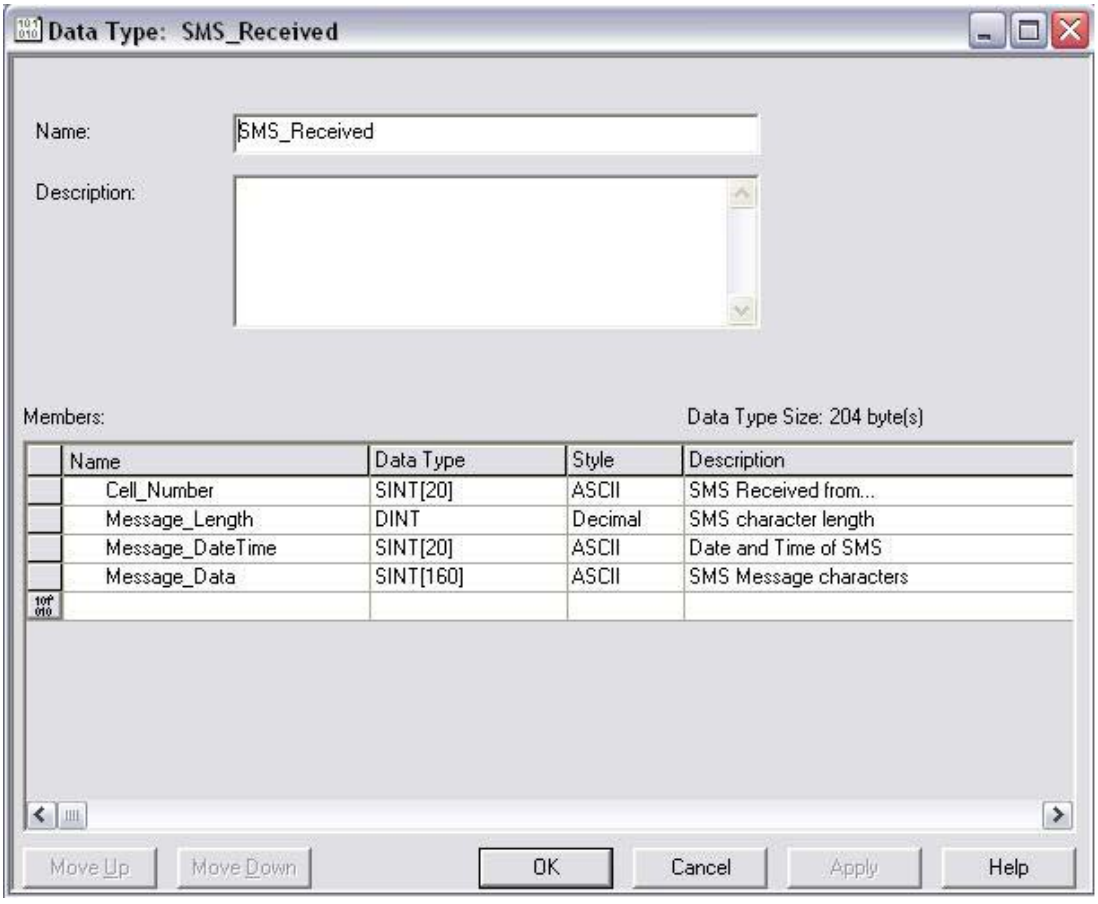


Figure 7.8.2: *SMS_Received UDT*

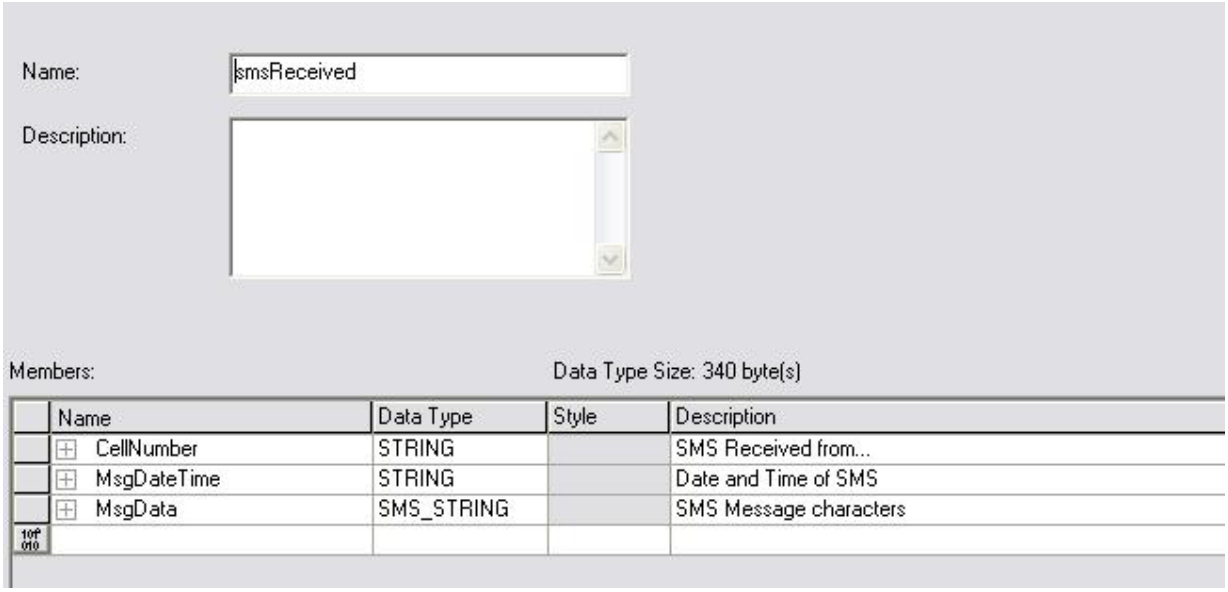


Figure 7.8.3: *Alternative SMS_Received UDT example*

In Figure 7.8.1, *SMS_List* is an array of SMS data types to save received SMS's with an increasing *SMSIndex* value.

In this example, the cell number and received message are copied into their respective positions in a SMS data type.

The message block is fired in the following manner based on the value of the *LastSMSCount variable:*

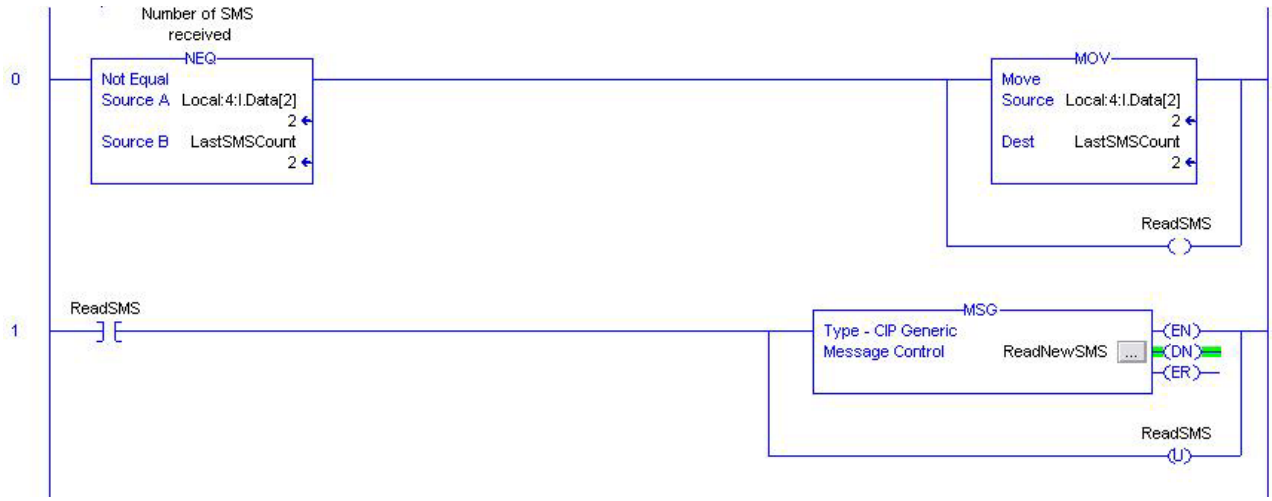


Figure 7.8.4: *Receiving Message Block firing*

The following is a list of the parameters needed for sending and receiving SMS's (using the sample code provided):

Name	Value	Force Mask	Style	Data Type
ConfigSMS	{...}	{...}		GSM_Config
LastSMSCount	0		Decimal	DINT
Local:4:C	{...}	{...}		AB:1756_MODULE:C:0
Local:4:I	{...}	{...}		AB:1756_MODULE_DINT_...
Local:4:O	{...}	{...}		AB:1756_MODULE_DINT_...
ReadNewSMS	{...}	{...}		MESSAGE
ReadSMS	0		Decimal	BOOL
Response	{...}	{...}	Decimal	DINT[50]
SendSMSMsg	{...}	{...}		MESSAGE
SMS_List	{...}	{...}		SMS_Received[50]
SMS_Response	{...}	{...}	Decimal	INT[150]
SMSIndex	0		Decimal	DINT
smsONS	0		Decimal	BOOL
SMSSendReq	{...}	{...}		SMS_Sending
smsSentCount	0		Decimal	DINT
smsTimer	{...}	{...}		TIMER
StoreMessageONS	0		Decimal	BOOL

Figure 7.8.5: *Parameter List*

The following ladder rungs can be added to the "Main" routine – These load the Configuration image and fire the Send/Receive routines when applicable:

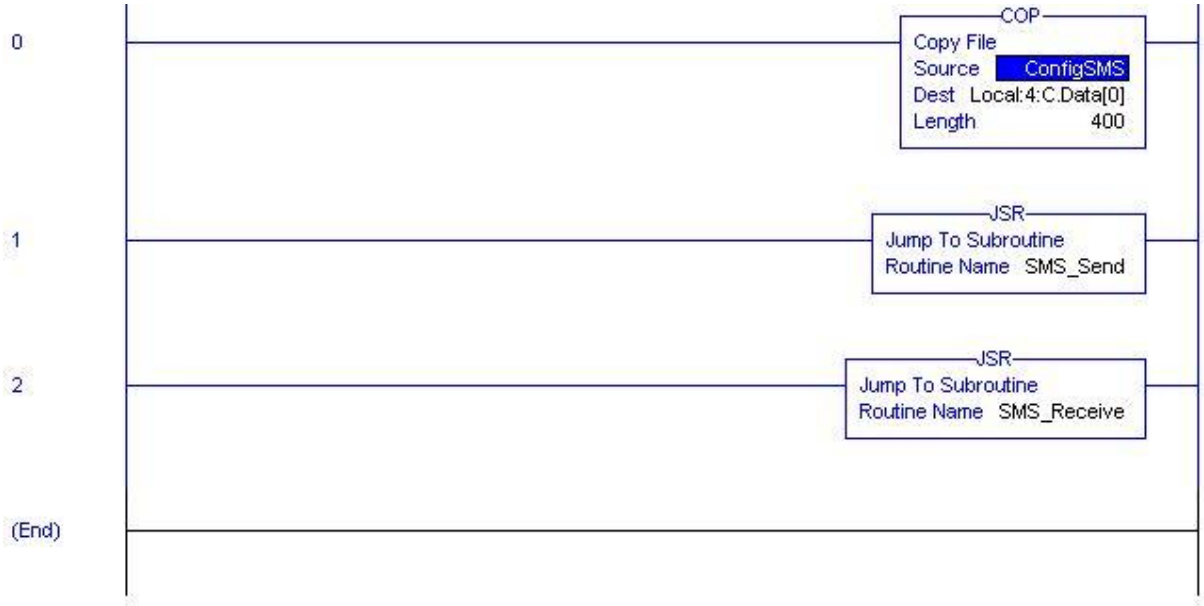


Figure 7.8.6: *Main Routine ladder*

CHAPTER 8 GPRS COMMUNICATION

GPRS Communication is accomplished by setting up a CIP Generic Custom message block.

The 1756HP-GSM Master module is designed in such a way that it can establish a one-to-one GPRS connection with multiple slave modules. The Master connects to a slave, sends a request, receives the requested data, and then disconnects. The process of contacting one slave takes approximately 15 – 30 seconds under good signal conditions. This is dependent on network availability and good network conditions.

The Master can contact multiple slaves in a “polling” fashion. Each slave is required to have a unique SIM card with a network-assigned unique IP address.

GPRS communication is very cost effective and is calculated by the volume of data sent and received by each module, and not the time you spend connected.

The GPRS communication uses a communication protocol called CSpeak. The following data characteristics are associated with this protocol:

	Minimum Packet Size
Master Requests:	
Request Read OutputImage	**14 Bytes**
Request Write InputImage	**12 Bytes + Payload**
Slave Messages:	
Response Read OutputImage	**12 Bytes + Payload**
Response Write InputImage (Ack)	**14 Bytes**

Data Cost Examples:

Using an example Network Supplied SIM contract with the following details:

Contract: R39-00 /month (Data included 5 Megabyte)
Effective In-bundle rate R7.80 per MB
Out-of-bundle rate R2-00 per MB
(Approx. $7.8x10^{-6}$ cents per byte)

Scenario 1

ONE 1756HP-GSM Master
ONE 1756HP-GSM Slave

a) We are sending a *Request Read OutputImage* ONCE every 10 minutes and we are requesting 10 Integers (16bit). (10 integers = 20 bytes)

SIM Total:
Per Day: 144 requests Sent + 144 messages Received

a) (144 x 14bytes) + (144 x (12+20)bytes) = 6624 bytes

Total for Master = **6624 bytes per day**

Total for Slave = **6624 bytes per day**

Total: **410688 bytes** per month (31 days) per SIM card

Total: **R0.50 / month** per SIM card

Note:
The Total bytes for the month is less than the data included in the contract therefore the Monthly cost will remain at **R39-00 / month per SIM card.**

Scenario 2

a)
ONE 1756HP-GSM Master
FIVE 1756HP-GSM Slaves

AND

b)
ONE 1756HP-GSM Master
FIVE 1756HP-GSM Slaves

Each Master Communicating with 5 slaves (polling one at a time).

a) We are *Request Read OutputImage* ONCE every 5 minutes from all 5 slaves of 50 Integers (100 bytes).
b) We are sending a *Request Write InputImage* ONCE every 10 minutes to the other 5 slaves of 10 integers (20 bytes).

a)
SIM Total (per slave):
Per Day: ((288 requests received) + (288 messages sent))
= (288 x 14bytes) + (288 x (12+100bytes))
= 4032 + 32256 = 36288 bytes.

SIM Total (master):
Per Day: 36288 x 5 = 181440 bytes.

Total per day: 217728 bytes / day

Total per month: 6749568 bytes (approx 6.7MB / month)

Each Slave uses: 1124928 bytes (approx 1.1MB) per month

Master uses: 5624640 bytes (approx 5.6MB) per month

b)
SIM Total (per slave):
Per Day: ((144 *Write InputImage Received*) + (144 *Acks* sent))
= (144 x (12+20bytes)) + (144 x 14) = 6624 bytes.

SIM Total (master):
Per Day: 6624 x 5 = 33120 bytes.

Each Slave uses: 205344 bytes (less than 0.5MB) per month

Master uses: 1026720 bytes (approx 1MB) per month

Message Block Configuration:

The configuration of the GPRS message instruction is as follows:

Field	Value
Message Type	CIP Generic
Service Type	Custom
Service Code	0x32
Class	0x71
Instance	0x02
Attribute	0x01
Source Length	232 bytes
Destination Element	Destination tag for reply data

***Table 8.1.1**: GRPS configuration.*

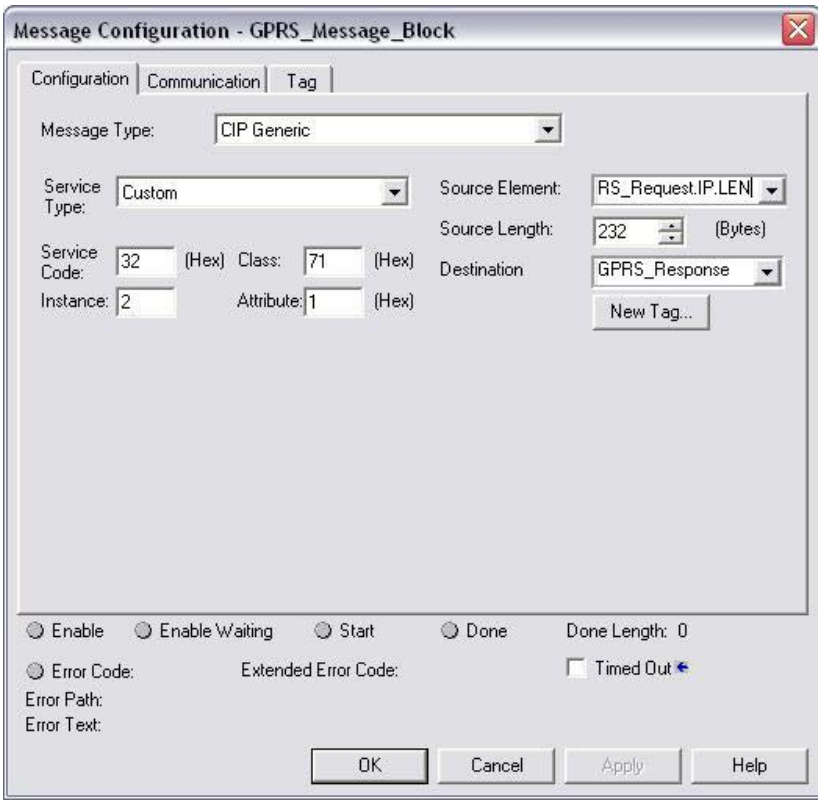


***Figure 8.1.a:** GPRS Message Block Configuration*

GPRS_Message_Block	{...}	{...}		MESSAGE	
GPRS_Message_Block.Flags	16#0200		Hex	INT	
GPRS_Message_Block.EW	0		Decimal	BOOL	
GPRS_Message_Block.ER	0		Decimal	BOOL	
GPRS_Message_Block.DN	0		Decimal	BOOL	
GPRS_Message_Block.ST	0		Decimal	BOOL	
GPRS_Message_Block.EN	0		Decimal	BOOL	
GPRS_Message_Block.TO	0		Decimal	BOOL	
GPRS_Message_Block.EN_CC	1		Decimal	BOOL	
GPRS_Message_Block.ERR	16#0000		Hex	INT	
GPRS_Message_Block.EXERR	16#0000_0000		Hex	DINT	
GPRS_Message_Block.ERR_SRC	0		Decimal	SINT	
GPRS_Message_Block.DN_LEN	0		Decimal	INT	
GPRS_Message_Block.REQ_LEN	232		Decimal	INT	
GPRS_Message_Block.DestinationLink	0		Decimal	INT	
GPRS_Message_Block.DestinationNode	8#000_000		Octal	INT	
GPRS_Message_Block.SourceLink	0		Decimal	INT	
GPRS_Message_Block.Class	16#0071		Hex	INT	
GPRS_Message_Block.Attribute	16#0001		Hex	INT	
GPRS_Message_Block.Instance	2		Decimal	DINT	
GPRS_Message_Block.LocalIndex	0		Decimal	DINT	
GPRS_Message_Block.Channel	'$00'		ASCII	SINT	
GPRS_Message_Block.Rack	8#000		Octal	SINT	
GPRS_Message_Block.Group	0		Decimal	SINT	
GPRS_Message_Block.Slot	0		Decimal	SINT	
GPRS_Message_Block.Path	'$01$05'	{...}		STRING	
GPRS_Message_Block.RemoteIndex	0		Decimal	DINT	
GPRS_Message_Block.RemoteElement	''	{...}		STRING	
GPRS_Message_Block.UnconnectedTimeout	60000000		Decimal	DINT	NOTE THIS VALUE MUST BE THE SAME
GPRS_Message_Block.ConnectionRate	7500000		Decimal	DINT	
GPRS_Message_Block.TimeoutMultiplier	0		Decimal	SINT	

Figure 8.1.b: *GPRS Message Block Configuration Values*

Note: The Value of the Unconnected Message Block Timeout must be set to between 60seconds and 90seconds. Ie. 60000000 = 60s

Creating a Connection:

To create a data transfer connection, we need a GPRS Message Block and a variable of the UDT *GPRS_Connect_Request* (called GPRS_Request in this example Fig 8.2 and 8.3).

Using this variable we specify the IP address and Port Number we want to connect to as well as the *Function Number* we would like to use (with their respective other reference values).

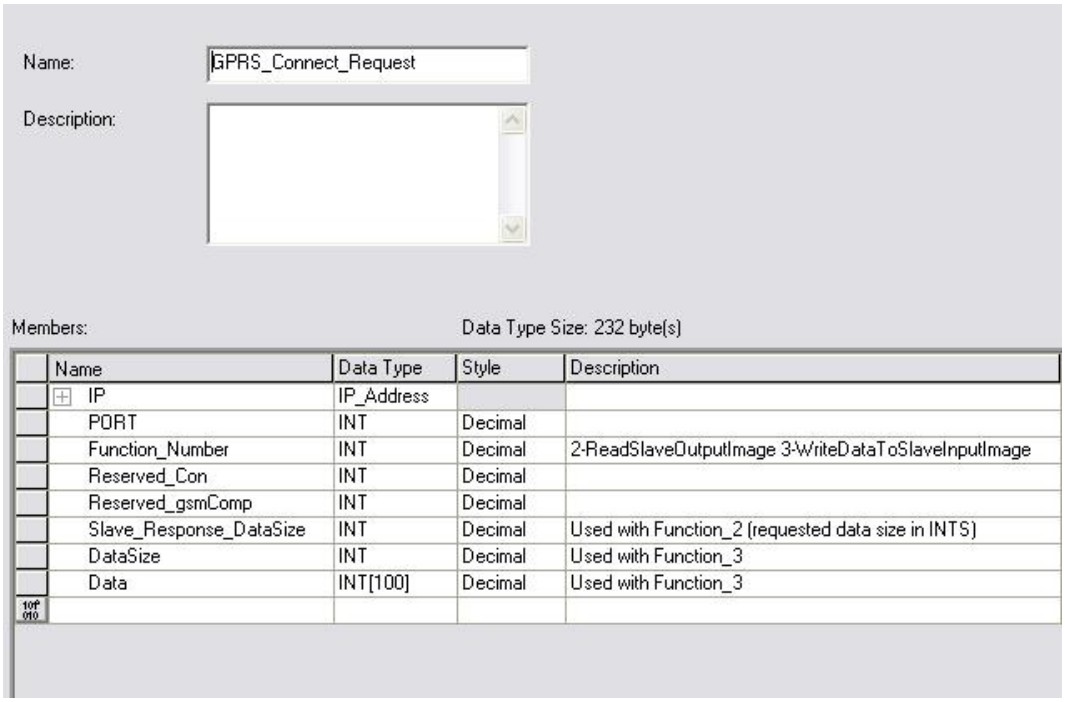
The IP address is the address associated with the SIM card (obtained through an APN)

The Port number can range from 0…65535

Function Numbers:

#2: Read the Slave's OutputImage[size] (returns values to response array)
Slave_Response_DataSize: Number of Integers required from slave.

#3: Write Data to Slave's InputImage (data is sent to slave module)
DataSize: Number of Integers to send to slave.
Data[0…99]: Physical data to send to slave.



Name	Data Type	Style	Description
IP	IP_Address		
PORT	INT	Decimal	
Function_Number	INT	Decimal	2-ReadSlaveOutputImage 3-WriteDataToSlaveInputImage
Reserved_Con	INT	Decimal	
Reserved_gsmComp	INT	Decimal	
Slave_Response_DataSize	INT	Decimal	Used with Function_2 (requested data size in INTS)
DataSize	INT	Decimal	Used with Function_3
Data	INT[100]	Decimal	Used with Function_3

Figure 8.2: *GPRS Connect Request UDT*

GPRS_Request	{...}	{...}		GPRS_Connect_Request	
GPRS_Request.IP	'196.135.148.3'	{...}		IP_Address	
GPRS_Request.PORT	5050		Decimal	INT	
GPRS_Request.Function_Number	2		Decimal	INT	2-ReadSlaveOutputImage 3-WriteDataToSlaveInputImage
GPRS_Request.Reserved_Con	0		Decimal	INT	
GPRS_Request.Reserved_gsmC...	0		Decimal	INT	
GPRS_Request.Slave_Response...	100		Decimal	INT	Used with Function_2 (requested data size in INTS)
GPRS_Request.DataSize	100		Decimal	INT	Used with Function_3
GPRS_Request.Data	{...}	{...}	Decimal	INT[100]	Used with Function_3
GPRS_Request.Data[0]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[1]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[2]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[3]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[4]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[5]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[6]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[7]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[8]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[9]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[10]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[11]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[12]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[13]	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[14]	0		Decimal	INT	Used with Function_3
...	0		Decimal	INT	Used with Function_3
...	0		Decimal	INT	Used with Function_3
...	0		Decimal	INT	Used with Function_3
GPRS_Request.Data[99]	0		Decimal	INT	Used with Function_3

Figure 8.3: *GPRS_Request variable*

Function (2) will return the data into the destination response array with the following format:

Field	Bytes	Type	Description
Response[0]	2	INT	Number of INTS requested.
Response[1]	2	INT	Slave's OutputImage[0]
…..			
Response[n]	2	INT	Slave's OutputImage[n-1]

where n indicates the number of INTs requested.

Table 8.1.2*: GPRS Read OutputImage response.*

The **Datasize** of the GPRS payload it restricted to:

Minimum: 1 INT (2 bytes)
Maximum: 100 INTs (200 bytes) per message.

Function (3) returns an Acknowledgement and therefore no data will be loaded into the response array.

The data that is sent to the slave is loaded into its InputImage starting at index position 12 (see Fig 8.4)

Note:

Local:s:I:Data[12] constitutes two INTs:

HIGH-INT : Rx Packet TNS Number
LOW-INT : Number of INTS that have been received.

In Figure 8.4: Local:s:I:Data[12] = 16#0002_000a

This means:

TNS value is 2 (Three Function3 packets have arrived: 0,1 and 2)
The data size is 10 INTS (0x000a)

Data Received: Integer1: 0x0005 : (decimal: 5)
Integer2: 0x0006 : (decimal: 6)
Integer3: 0x0007 : (decimal: 7)
Integer4: 0x0008 : (decimal: 8)
Integer4: 0x0009 : (decimal: 9)
Integer4: 0x000a : (decimal: 10)

Name	Value	Force	Style	Data Type
Local:4:I.Data	{...}	{...}	Decimal	DINT[63]
Local:4:I.Data[0]	2#0000_0000_0000_0000_0000_0100_1100_0101		Binary	DINT
Local:4:I.Data[1]	74		Decimal	DINT
Local:4:I.Data[2]	0		Decimal	DINT
Local:4:I.Data[3]	0		Decimal	DINT
Local:4:I.Data[4]	16#c487_9403		Hex	DINT
Local:4:I.Data[5]	16#c487_940a		Hex	DINT
Local:4:I.Data[6]	0		Decimal	DINT
Local:4:I.Data[7]	0		Decimal	DINT
Local:4:I.Data[8]	16#0000_0002		Hex	DINT
Local:4:I.Data[9]	0		Decimal	DINT
Local:4:I.Data[10]	0		Decimal	DINT
Local:4:I.Data[11]	0		Decimal	DINT
Local:4:I.Data[12]	16#0002_000a		Hex	DINT
Local:4:I.Data[13]	16#0006_0005		Hex	DINT
Local:4:I.Data[14]	16#0008_0007		Hex	DINT
Local:4:I.Data[15]	16#000a_0009		Hex	DINT
Local:4:I.Data[16]	0		Decimal	DINT
Local:4:I.Data[17]	0		Decimal	DINT

Figure 8.4: *InputImage of GSM Module*

The IP_Address data type is defined as:

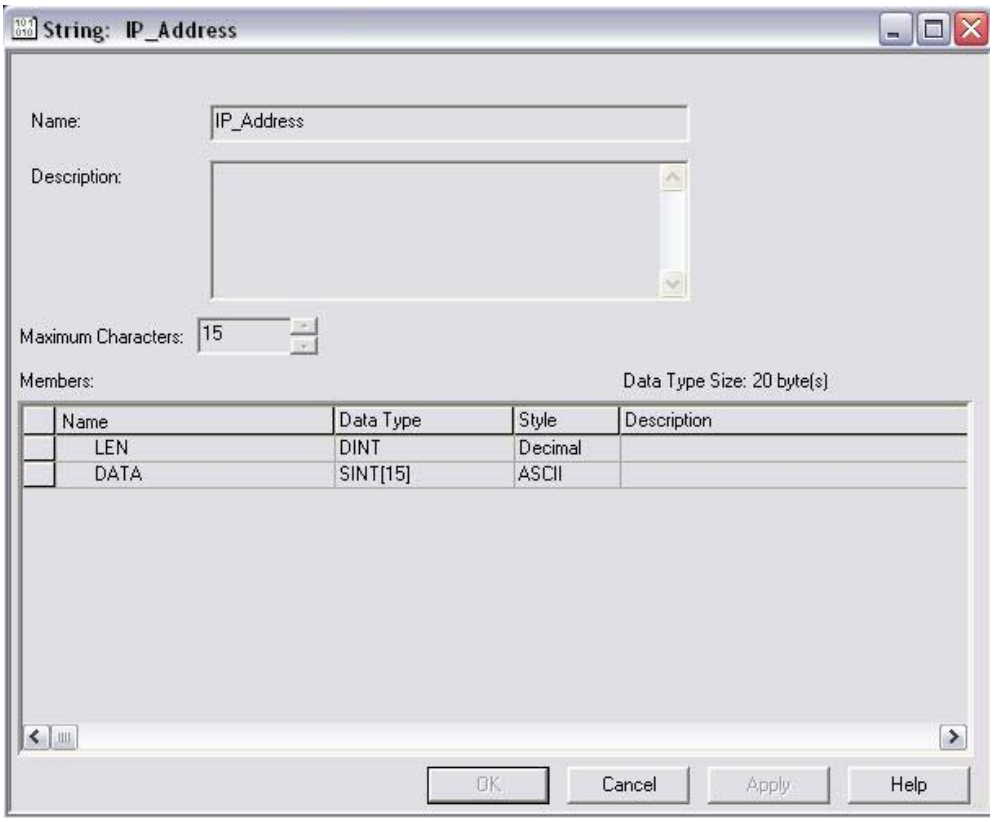


Figure 8.5: *IP_Address Data Type*

The **IP_Address** field will only accept an IP address in the correct format. The format is

[**Byte1**] [**.**] [**Byte2**] [**.**] [**Byte3**] [**.**] [**Byte4**]

Example: **192.168.10.66**

Note: If the IP address is in the incorrect format, the message block will Fail with Error Code: 16#0003

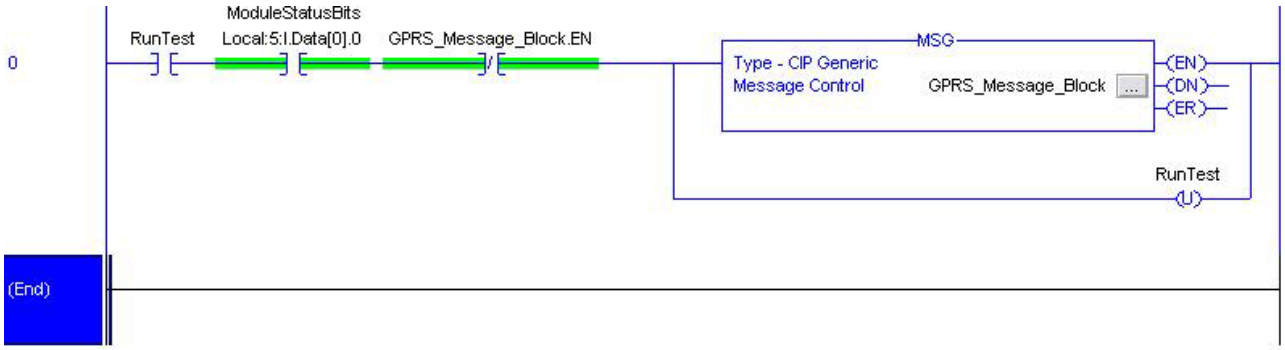


Figure 8.6: *Read OutputImage Message ladder example*

Network Examples:

Eg1: SMS

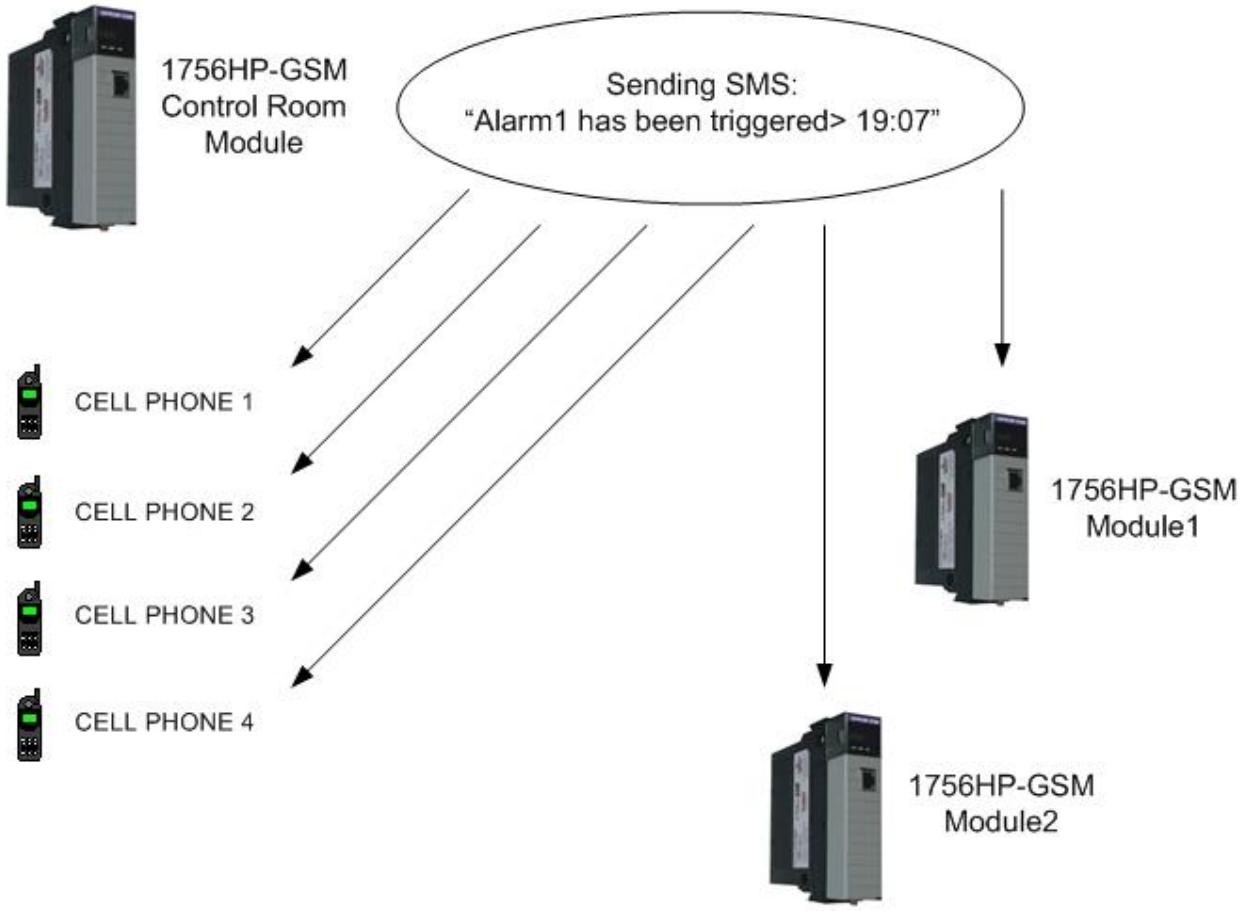
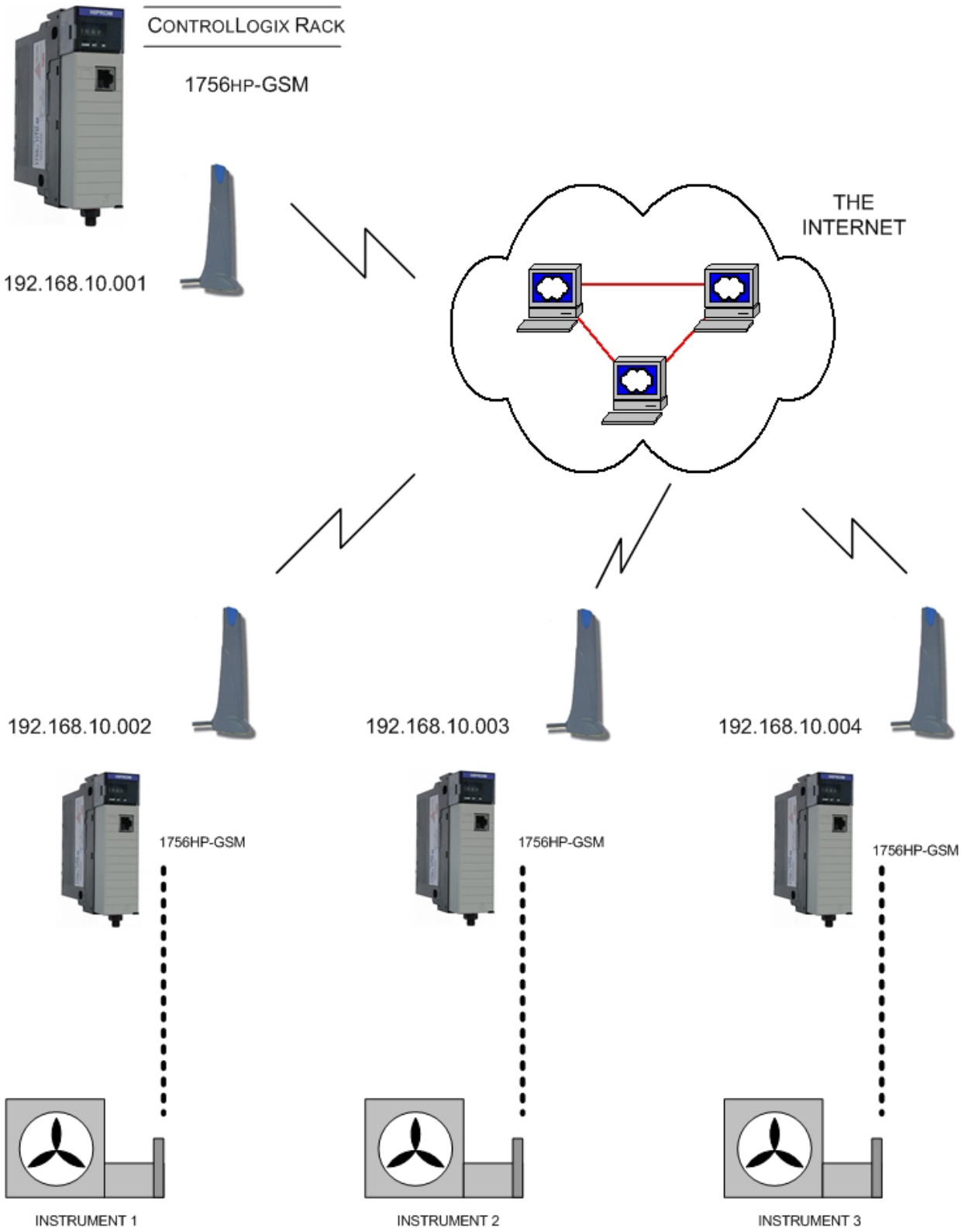


Figure 8.7: *SMS Network Example*

Eg2: GPRS


CONTROLLOGIX RACK
1756HP-GSM
192.168.10.001
THE INTERNET
192.168.10.002
192.168.10.003
192.168.10.004
1756HP-GSM
1756HP-GSM
1756HP-GSM
INSTRUMENT 1
INSTRUMENT 2
INSTRUMENT 3

APPENDIX A MODULE STATUS

The following sections describe the status indicators of the module:

A.1 Status LEDs of Display

DESCRIPTION	STATUS	MEANING
Module Status	Solid Red	Initialization or Watch Dog Fault
	Flashing Red	Major Fault
	Flashing Green	Minor Fault
	Green	Module operating correctly

A.2 GSM Status LEDs (Located on the bottom of the module)

LED	DESCRIPTION	STATUS	MEANING
RED	Power	Solid Red	The GSM module is powered, and functioning properly
		Off	No power is being supplied to the GSM module, the module is not operating correctly
GREEN	Registration Status	Flashing Quickly (Approx. Every 1 Sec)	The Module has not established connection with the GSM network (Approx. every 1 Sec)
		Flashing Slowly (Approx. Every 3 Sec)	The module has successfully established a connection with the GSM network.

A.3 Status Display

Init	**Initialization of Module** The module is initialized only on power-up.
Frn	**Firmware Revision** The firmware revision number is displayed on power-up.
Conn	**Establishing Connection** The module is attempting to connect to the network.
PnOK	The PIN code has been accepted.
PINx	The PIN code has not been accepted.
PUK	**PUK Required** The SIM card requires the PUK code.
Err	**Error Received** An error has been received; see ERROR CODE in the InputImage.
GSM	Module is configured as a standard GSM module. GPRS is disabled.
GPRS	Module is configured as a GPRS module
SMSr	Reading new received SMS from the module
SMSt	Transmitting an SMS from the module
SSxx	**Signal Strength** Xx is the GSM network signal strength expressed in percentage
conA	**Connection Attempt** Master module is attempting to establish a connection with a slave module.
conS	**Connection Success** GPRS connection has been successfully established, data communication will now follow assuming sufficient signal quality.
conE	**Connection Failure** GPRS connection has failed. See extended error code.
conR	**Connection Received** A successful GPRS connection has been received from a master module.
RptC	**Connection Retry** Module is retrying the connection attempt.
Dsc√	**Disconnect Success** Module has successfully Disconnected.
CSP√	**Packet Received** A CSpeak/GPRS data packet has been received.
Ack√	**Acknowledgement Received** A CSpeak/GPRS acknowledgment has been received.
RtCA	**Context Retry** Module is attempting to reset GPRS context.
Snt√	**SMS Message Sent**

A.4 Local Error Codes

These error codes are located in **Local:s:I.Data[0].16..23** as an SINT value. The error code will occasionally be echoed in the Extended error code DINT value (**Local:s:I.Data[6])**

Error Code	Description	LED Display (if any)	Extended Error Code	MSG Block Error Code (if any)
10	**SIM not inserted**	**NoSM**		
11	SIM PIN required	SimE		
12	SIM PUK required	PUK		
13	SIM failure	SimE		
14	SIM busy	SimE		
15	SIM wrong	SimE		
16	**SIM Code Incorrect**	**PINx**		
17	SIM PIN2 required	PN2x		
18	SIM PUK2 required	PUK2		
20	memory full	SimE		
23	memory failure	SimE		
30	**no network service / no balance on Sim**	**AIRx**		
31	network timeout	SimE		
82	Timeout in opening socket		409	6
83	Connection failed		412	7
84	Context not opened		406	8
85	TX error		413	
87	Activation failed		405	
98	Already Listening (No Err)		414	
	CRC Error		399	1

A.5 Message Block Extended Error Codes (SMS)

Code	Description
16#0000 0204	Message Block Timeout
16#0000 0002	SMS timeout error. (potential network problem)
16#0000 0003	SMS Sending error. Cell number and/or message incorrect format)

A.6 Message Block Extended Error Codes (GPRS)

Code	Description
16#0000 0204	Message Block Timeout
16#0000 0001	CRC Error
16#0000 0002	GPRS Response has not been received correctly (potential signal / network problem)
16#0000 0003	Request IP address incorrect format / length
16#0000 0004	Module is in wrong Mode.
16#0000 0006	Timeout in opening Socket. Slave might not be online. Network quality / antenna status might be suspect. Retry connect.
16#0000 0007	Connection failed. Slave might not be online / Incorrect IP / Incorrect Port . Retry connect.
16#0000 0008	Context Not Opened. Module will try to establish context. Retry connect.
16#0000 000B	Internal Network Recover.

APPENDIX B SPECIFICATIONS

Parameter	Specification
General	
Module Location	Any Slot
Electrical	
Backplane Current	515mA @ 5.1V
	3mA @ 24V
Schedules Connection Parameters	
RPI	5ms to 750ms
GPRS Specification	
GSM Module:	Telit GE863-QUAD
	RoHS Compliant
	Quad Band
	GPRS Class 10
Max Average current consumption during GPRS transmission:	400mA
Max Peak current consumption (impulsive):	1.9A
Antenna (GSM Shark Fin)	
Antenna Connector	SMA male connector
Frequency Range	824 - 960 MHz and 1700 - 2170 MHz
Polarization	Linear (Vertical or Horizontal)
Impedance	50Ω (nominal)
VSWR	< 2.5:1
Gain	1.5 dBi (± 0.5 dBi) (including cable losses)
Weight	300g